

Effectiveness and failure modes of error correcting code in industrial 65 nm CMOS SRAMs exposed to heavy ions*

TONG Teng (童腾),^{1,2} WANG Xiao-Hui (王晓辉),^{1,2} ZHANG Zhan-Gang (张战刚),^{1,2}
DING Peng-Cheng (丁朋程),^{1,3} LIU Jie (刘杰),¹ LIU Tian-Qi (刘天奇),^{1,2} and SU Hong (苏弘)^{1,†}

¹*Institute of Modern Physics, Chinese Academy of Science, Lanzhou, 730000, China*

²*University of Chinese Academy of Sciences, Beijing 10049, China*

³*Northwest Normal University, Lanzhou, 730000, China*

(Received July 23, 2013; accepted in revised form October 8, 2013; published online February 20, 2014)

Single event upsets (SEUs) induced by heavy ions were observed in 65 nm SRAMs to quantitatively evaluate the applicability and effectiveness of single-bit error correcting code (ECC) utilizing Hamming Code. The results show that the ECC did improve the performance dramatically, with the SEU cross sections of SRAMs with ECC being at the order of $10^{-11} \text{ cm}^2/\text{bit}$, two orders of magnitude higher than that without ECC (at the order of $10^{-9} \text{ cm}^2/\text{bit}$). Also, ineffectiveness of ECC module, including 1-, 2- and 3-bits errors in single word (not Multiple Bit Upsets), was detected. The ECC modules in SRAMs utilizing (12, 8) Hamming code would lose work when 2-bits upset accumulates in one codeword. Finally, the probabilities of failure modes involving 1-, 2- and 3-bits errors, were calculated at 39.39%, 37.88% and 22.73%, respectively, which agree well with the experimental results.

Keywords: Single event upsets (SEU), SRAM, Error correcting code (ECC), Hamming code, Effectiveness, Failure modes

DOI: [10.13538/j.1001-8042/nst.25.010405](https://doi.org/10.13538/j.1001-8042/nst.25.010405)

I. INTRODUCTION

As technology scales downward in modern integrated circuits, such as SRAM, the minimum charge needed to upset a device within a unit memory cell decreases, while the influence of charge sharing on adjacent unit memory cells increases [1–5]. Therefore, advanced devices (especially deep-submicrometer) are much more sensitive to the energy deposition in the device by heavy ion irradiation, and this critically restricts the devices' use in space.

Many methods have been proposed to mitigate the single event upsets (SEUs) occurred in advanced devices. Bits interleaving architecture is a commonly accepted approach to mitigate Multiple Bit Upsets (MBUs) in data word. In this architecture, the bits in a data word are not physically adjacent, but interleaved with bits of other data words. In this way, every MBU of physically adjacent memory cells is transformed into multiple single bit upsets (SBUs) in different memory words.

Error correcting code (ECC) utilizing Hamming code is found commonly in many high-reliability and performance applications. As a relatively simple yet powerful ECC code, it corrects single bit errors anywhere within the codeword.

Therefore, MBUs which is now a major reliability problem in commercial and industrial electronics, can be transformed into multiple SBUs appear to be uncorrelated events relative to the ECC algorithm, and then be corrected [2, 5–7].

In this hardening approach, ECC module can be used in high-reliability and performance applications to resolve SBUs combining with the bits interleaving architecture in advanced process node devices.

To observe and compare the SEUs induced by heavy ions in SRAMs of different process, and to quantitatively evaluate the applicability and effectiveness of single-bit ECC utilizing Hamming code in advanced process SRAMs, we used ^{12}C ion beam to irradiate four SRAMs from ISSI company. Two of them, manufactured via 130 nm and 150 nm process, are the most advanced process devices in their SRAMs without ECC module, while the other two are of 65 nm process SRAMs with ECC module. Some interesting results were obtained.

II. EXPERIMENTAL BACKGROUND

Four industrial SRAMs, produced by high-performance CMOS technology, were irradiated at normal incidence in the vacuum by ^{12}C beams from the Heavy Ion Research Facility in Lanzhou (HIRFL). The ^{12}C ions were of effective linear energy transfer (LET) value of 1.8 MeV·cm²/mg. Table 1 shows the information of SRAMs under test. The IS2ME is 2 M-bit SRAM organized as 131 072 words by 16 bits with ECC, 65 nm process node; the IS4ME is 4 M-bit SRAM organized as 262 144 words by 16 bits with ECC, 65 nm process node; the IS2M is 2 M-bit SRAM organized as 131 072 words by 16 bits without ECC, 150 nm process node; and the IS4M is 4 M-bit SRAM organized as 262 144 words by 16 bits without ECC, 130 nm process node. The first two SRAMs with ECC are the main objects of observation, and the other two are the contrastive devices. All of the four industrial SRAMs belong to the IS61WV series made by ISSI company, and the ECC functions described in this application are made by Hamming code, a relatively simple yet powerful ECC which can correct all single bit errors in one codeword.

The SRAMs were tested using data pattern of all “1” (blanket pattern) at voltage of 3.3 V, and the work period was set at 20 MHz all the time. Under the static test mode, the devices were written prior to their beam-shot and read periodi-

* Supported by the National Natural Science Foundation of China (Nos. 11079045 and 11179003) and the Important Direction Project of the CAS Knowledge Innovation Program (No.KJCX2-YW-N27)

† Corresponding author, suhong@impcas.ac.cn

TABLE 1. The information of SRAMs under test

Device	Process node (nm)	Capacity (Mbits)	ECC	Abbreviation
61WV12816EDBLL-10TLI	65	2	with	IS2ME
61WV25616EDBLL-10TLI	65	4	with	IS4ME
61WV12816DBLL-10TLI	150	2	without	IS2M
61WV25616BLL-10TLI	130	4	without	IS4M

cally throughout the beam shot (this technique is often referred to as multiple-read) [1, 8, 9]. The error data occurred in the test were stored in another RAM (referred as mirrored RAM relative to the SRAM under test) working in the test system, as a referenced data for next read cycle. The test flow applied (Fig. 1) distinguishes SBU, MBU and SEL. All the upset events were recorded with a timestamp and bitmap location.

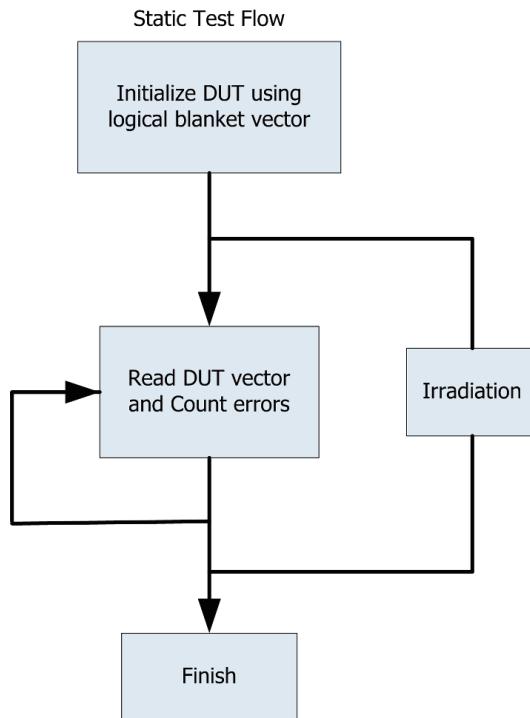


Fig. 1. Static test flow.

III. RESULT ANALYSIS AND DISCUSS

A. The high efficiency of ECC module

SEU cross sections of the four SRAMs are shown in Fig. 2. One sees that the SRAMs without ECC module are much more sensitive to the irradiation than the devices with ECC module. The SEU cross sections of SRAMs without ECC module are at the order of $10^{-9} \text{ cm}^2/\text{bit}$, while they are $10^{-11} \text{ cm}^2/\text{bit}$ for SRAMs with ECC module. However, the technology of producing the IS2ME and IS4ME in 65 nm process is more advanced than IS2M (150 nm process node) and IS4M (130 nm process node). With technology scaling, the number of upsets per chip increases due to higher circuit density and sensitivity.

Therefore, the sharp contrast of the two datum groups should be attributed to the high efficiency of ECC module.

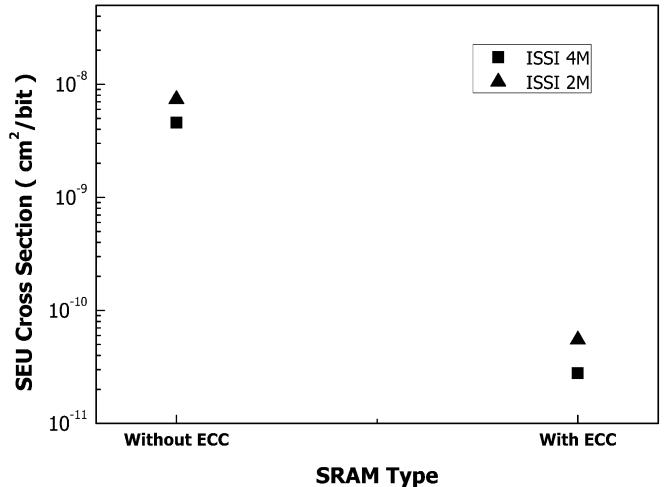


Fig. 2. Cross sections of four SRAMs.

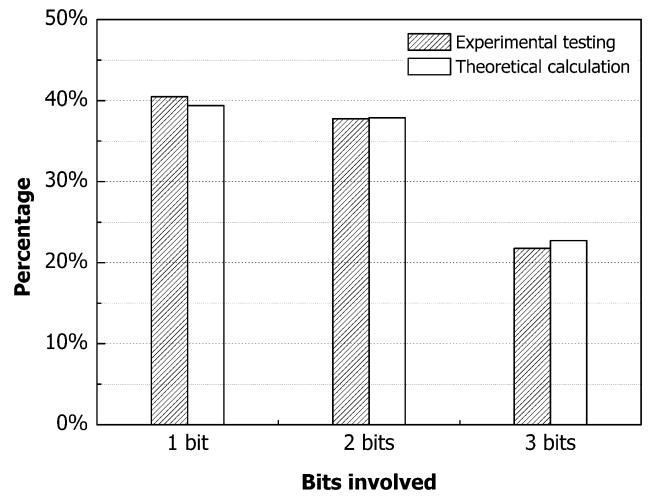


Fig. 3. The bits per upset event distribution.

B. The ineffectiveness of ECC module

Only 1 bit upsets in a data word were detected in the devices without ECC module in this experiment. The upset events involving 1, 2 and 3 bits errors occurred in the devices with ECC

module. Fig. 3 shows the measured and theoretical results of bits per upset event distribution (percentages over total events). We will discuss the results with an emphasis: special attentions shall be paid to the word “upset” and “error” in the following text—“upset” is the real change occurred in memory cell, and “error” is the data being read out from the memory finally.

1. The fundamental reason

For discussing the experimental results, we have the following assumptions:

1. Considering the beam energy of ^{12}C and the bits interleaving architecture, the normal incidence ion beams do not affect the adjacent memory cells simultaneously. So, MBUs are not supposed to occur in a codeword any time in this experiment [2, 5–7].
2. The static mode used in this test means that only one write operation worked in a test cycle, while the ECC module does not correct or re-write the memory itself [1], but just corrects the “error” bit(s). When the data be read out through ECC module, the memory remains in upset status until a new write command arrives with new data. Therefore, if other bit(s) upset occurs in the same word, the ECC module utilizing Hamming code, which can only correct one bit error, will lose function. So, the disablement of ECC module is an accumulation effect caused by several SBUs in a word at different time. On the other hand, as ECC functional block diagram (Fig. 4, presented in the datasheet of the devices with ECC module) shows, the circuit structure of ECC module utilizes the (12, 8) Hamming code in the application.

Based on time structure of the cyclotron and the upstream scanning magnets, the incident ions are of uniform temporal and spatial distribution in the used flux range, thus each SBU could be deemed as an independent random event.

In independent random event, if the upset probability is p ($p \ll 1$), the probability that r bit(s) upset occurs in an n bits codeword is $P_n(r) = C_n^r p^r (1-p)^{n-r} \approx \frac{n!}{r!(n-r)!} p^r$. From the results of IS2M and IS4M, about 200 ions could cause 1 bit upset in order of magnitude, assuming this probability is suitable for IS2ME and IS4ME, we have $p = 5 \times 10^{-3}$. Then, the probability of two and three SBUs occurring at different time in one codeword is

$$\begin{aligned} P_{12}(2) &= C_{12}^2 p^2 (1-p)^{12-2} \\ &\approx \frac{12!}{2!(12-2)!} (5 \times 10^{-3})^2 \\ &= 3.3 \times 10^{-4}. \end{aligned} \quad (1)$$

The probability of three SBUs occurs in different time in one codeword is

$$\begin{aligned} P_{12}(3) &= C_{12}^3 p^3 (1-p)^{12-3} \\ &\approx \frac{12!}{3!(12-3)!} (5 \times 10^{-3})^3 \\ &= 1.1 \times 10^{-6}. \end{aligned} \quad (2)$$

The results of Eq. (1) and Eq. (2) show a probability difference of two orders of magnitude between $r = 2$ and $r = 3$. Thus three or more SBUs occur at different time in one codeword is of very low probability, hence their omission in this experiment.

Therefore, the fundamental reason for the problem is that a 2 bits upset in a codeword causes the disablement of ECC module utilizing (12, 8) Hamming code.

2. Parsing the problem

Figure 5 is a basic memory architecture of ECC module utilizing Hamming code [10]. Table 2 is a common relationship between syndrome vector and single-error location.

TABLE 2. The relationship between syndrome vector and single-error location

$S_3S_2S_1S_0$	Error location	$S_3S_2S_1S_0$	Error location
0001	P_0	1000	P_3
0010	P_1	1001	D_4
0011	D_0	1010	D_5
0100	P_2	1011	D_6
0101	D_1	1100	D_7
0110	D_2	—	—
0111	D_3	0000	No error

Assuming the 12 bits codeword is $D_7D_6D_5D_4P_3D_3D_2D_1P_2D_0P_1P_0$, 8 bits data word is vector \mathbf{D} and 4 bits check word is vector \mathbf{P} , the syndrome vector \mathbf{S} can be generated by data word and check word as [11]:

$$\begin{aligned} S_0 &= D_0 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus P_0; \\ S_1 &= D_0 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 \oplus P_1; \\ S_2 &= D_1 \oplus D_2 \oplus D_3 \oplus D_7 \oplus P_2; \\ S_3 &= D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus P_3; \end{aligned} \quad (3)$$

means

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} D_7 \\ D_6 \\ D_5 \\ D_4 \\ P_3 \\ D_3 \\ D_2 \\ D_1 \\ P_2 \\ D_0 \\ P_1 \\ P_0 \end{bmatrix}, \quad (4)$$

and the corresponding (12, 8) parity matrix is

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

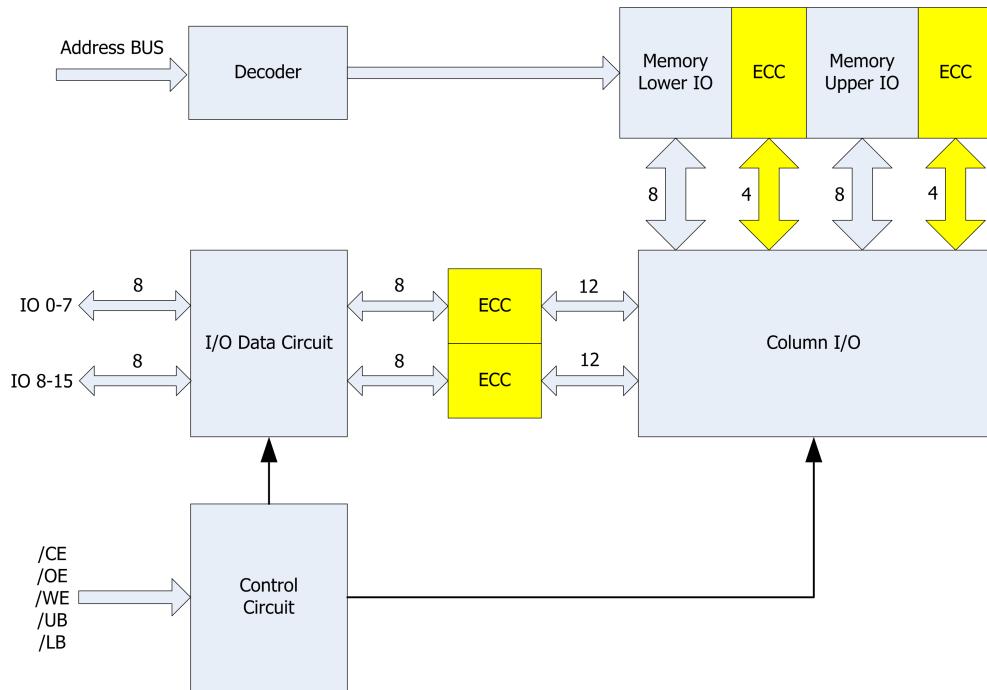


Fig. 4. Functional block diagram of SRAMs with ECC module.

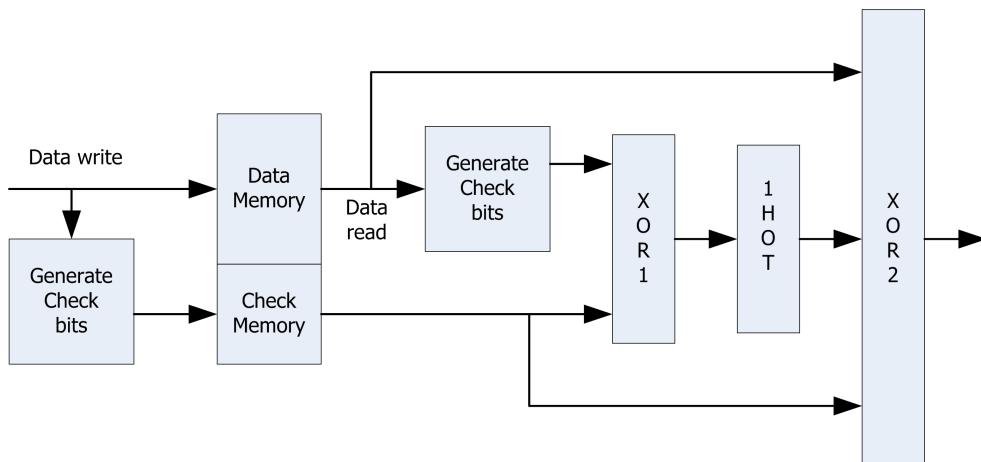


Fig. 5. Basic memory architecture for ECC module utilizing Hamming code.

When an 8-bits data word is written in SRAM, the ECC module generates a 4-bits check word to compose 12-bits codeword and store it in the memory cell. After irradiation, when the data word is read out from memory cell through the ECC module, which generates a syndrome vector $S = (S_3 S_2 S_1 S_0)$, according to the codeword.

In Eq. (5), each column vector in parity matrix represents the position of each bit ($D_u (u = 0, 1, \dots, 7)$ or $P_v (v = 0, 1, 2, 3)$) in the codeword, 0 means that the bit does not participate in the form of S_k ($k = 0, 1, 2, 3$), 1 means that the bit participates in the form of S_k ($k = 0, 1, 2, 3$). Then, how does the 2-bits change in codeword generate $S \neq (0000)$, and how does the S point to an error in Table 2? The method to find the failure modes is discussed as follows:

1. Neither the 2 upset bits participate in the S_k , $S_k = 0 \oplus 0 = 0$ to point to “no error”.
2. Both the 2 upset bits participate in the S_k , $S_k = 1 \oplus 1 = 0$ to point to “no error”.
3. Only one upset bit participates in the S_k , $S_k = 1 \oplus 0 = 1$ or $S_k = 0 \oplus 1 = 1$, the value of the corresponding S_k is always 1, so the ECC module spots an “error” and makes a “correct” operation.

Consequently, the S_k value is associated with the status of 2 upset bits participating in the S_k , and the relationship is a “XOR” operation between S_k and the 2 upset bits.

For example, if the 2-bits upset comes from $D_3 P_0$, they will not affect the value of S_0 (as both participate in it) and S_3 (as

neither participate in it). However, $S_1 = P_1 \oplus D_0 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6$ and $S_2 = P_2 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_7$ will result in $S = (S_3 S_2 S_1 S_0) = (0110)$, which can be understood simply as:

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \quad (6)$$

Eq. (6) points to an “error” position at D_2 by Table 2, then ECC module corrects the right value of D_2 to an error value, while the real upset bit D_3 is read out as an “right” data, leading a 2-bits errors as $D_3 D_2$. In other words, the data written in is “FF”, and the data read out is “F3” as an error to be detected.

Therefore, the problem-solving method can be simplified as the following procedures: 1) extract two columns vectors (2-bits upset occurring in the same bit in a codeword does not affect the S , hence the omission of this condition) from parity matrix of Eq. (5), 2) make an “XOR” operation with them as Eq. (6), 3) produce the syndrome vector S , 4) find the “error” position where S points to, 5) analyze the relationship between the “error” and the “upset”, and 6) a statistics of failure modes including 1-bit, 2-bits and 3-bits errors read out from the SRAMs can be achieved.

TABLE 3. Message of the failure modes of ECC module of a 2-bits upset with both upsets occurring in check word

Upset position	$S_3 S_2 - S_1 S_0$	“Error” position the S points to	Error read out	Error types (bits involved)
$P_1 P_0$	0011	D_0	D_0	1 bit
$P_2 P_0$	0101	D_1	D_1	1 bit
$P_3 P_0$	1001	D_4	D_4	1 bit
$P_2 P_1$	0110	D_2	D_2	1 bit
$P_3 P_1$	1010	D_5	D_5	1 bit
$P_3 P_2$	1100	D_7	D_7	1 bit

3. Analysis results

Extracting two columns of vector from parity matrix of Eq. (4), the total number of error types is $C_{12}^2 = 66$. Tables 3-5 list details of the failure modes and error types.

1. When 2-bits upset are both in check word (Table 3)

In this case, the ECC module makes a wrong operation, the number of error types is $C_4^2 = 6$, all the failure mode is 1-bit.

2. When 1 bit upset in check word, 1 bit upset in data word (Table 4)

In this case, the ECC module would makes a wrong operation, the number of error types is $C_4^1 C_8^1 = 32$, of which the number of 1-bit is 20, the number of 2-bits is 12, and the failure modes includes 1-bit and 2-bits.

3. When 2 bits upset are both in data word (Table 5)

In this case, the ECC module makes a wrong operation, the number of error types is $C_8^2 = 28$, of which the number of 2-bit is 13, and the number of 3-bit is 15, and the failure modes includes 2-bits and 3-bits.

TABLE 4. Message of the failure modes of ECC module with 1 bit upset in check word and 1 bit upset in data word

Upset Position	$S_3 S_2 - S_1 S_0$	“Error” position the S points to	Error read out	Error types (bits involved)
$D_0 P_0$	0010	P_1	D_0	1 bit
$D_1 P_0$	0100	P_2	D_1	1 bit
$D_2 P_0$	0111	D_3	$D_3 D_2$	2 bit
$D_3 P_0$	0110	D_2	$D_3 D_2$	2 bit
$D_4 P_0$	0001	P_3	D_4	1 bit
$D_5 P_0$	1011	D_6	$D_6 D_5$	2 bit
$D_6 P_0$	1010	D_5	$D_6 D_5$	2 bit
$D_7 P_0$	1101	no point	D_7	1 bit
$D_0 P_1$	0001	P_0	D_0	1 bit
$D_1 P_1$	0111	D_3	$D_3 D_1$	2 bit
$D_2 P_1$	0100	P_2	D_2	1 bit
$D_3 P_1$	0101	D_1	$D_3 D_1$	2 bit
$D_4 P_1$	1011	D_6	$D_6 D_4$	2 bit
$D_5 P_1$	1000	P_3	D_5	1 bit
$D_6 P_1$	1001	D_4	$D_6 D_4$	2 bit
$D_7 P_1$	1110	no point	D_7	1 bit
$D_0 P_2$	0111	D_3	$D_3 D_0$	2 bit
$D_1 P_2$	0001	P_0	D_1	1 bit
$D_2 P_2$	0010	P_1	D_2	1 bit
$D_3 P_2$	0011	D_0	$D_3 D_0$	2 bit
$D_4 P_2$	1101	no point	D_4	1 bit
$D_5 P_2$	1110	no point	D_5	1 bit
$D_6 P_2$	1111	no point	D_6	1 bit
$D_7 P_2$	1000	P_3	D_7	1 bit
$D_0 P_3$	1011	D_6	$D_6 D_0$	2 bit
$D_1 P_3$	1101	no point	D_1	1 bit
$D_2 P_3$	1110	no point	D_2	1 bit
$D_3 P_3$	1111	no point	D_3	1 bit
$D_4 P_3$	0001	P_0	D_4	1 bit
$D_5 P_3$	0010	P_1	D_5	1 bit
$D_6 P_3$	0011	D_0	$D_6 D_0$	2 bit
$D_7 P_3$	0100	P_2	D_7	1 bit

Therefore, the total number of 1-bit is $6 + 20 = 26$, the probability in all error types is $26/66 = 39.39\%$; the total number of 2-bits is $12 + 13 = 25$, the probability in all error types is $25/66 = 37.88\%$; and the total number of 3-bits is 15, the probability in all error types is $15/66 = 22.73\%$. Table 6 shows the theoretical probabilities of failure modes including 1-, 2- and 3-bits agree well with the experimental results.

Therefore, the imminent factor of failure modes of ECC module in this experiment is due to the failure of (12, 8) Hamming code facing to 2 bits upset in one codeword.

IV. CONCLUSION

The results show the effectiveness and ineffectiveness of ECC module utilizing (12, 8) Hamming code in 65 nm process node SRAMs. The ECC module works obviously in hardening the advanced process node SRAMs. The failure modes including 1-, 2-, and 3-bits in a data word has been analyzed, and the essential factor of failure modes is due to the failure of (12, 8) Hamming code facing to 2 bits upset in one codeword. The measured bits per upset event distribution agree well with the theoretical calculation.

TABLE 5. The message of the failure modes of ECC module when 2 bits upset occur both in data word

Upset position	S ₃ S ₂ S ₁ S ₀	“Error” position the S points to	Error read out	Error types (bits involved)
D ₁ D ₀	0110	D ₂	D ₂ D ₁ D ₀	3 bit
D ₂ D ₀	0101	D ₁	D ₂ D ₁ D ₀	3 bit
D ₃ D ₀	0100	P ₂	D ₃ D ₀	2 bit
D ₄ D ₀	1010	D ₅	D ₅ D ₄ D ₀	3 bit
D ₅ D ₀	1001	D ₄	D ₅ D ₄ D ₀	3 bit
D ₆ D ₀	1000	P ₃	D ₆ D ₀	2 bit
D ₇ D ₀	1111	no point	D ₇ D ₀	2 bit
D ₂ D ₁	0011	D ₀	D ₂ D ₁ D ₀	3 bit
D ₃ D ₁	0010	P ₁	D ₃ D ₁	2 bit
D ₄ D ₁	1100	D ₇	D ₇ D ₄ D ₁	3 bit
D ₅ D ₁	1111	no point	D ₅ D ₁	2 bit
D ₆ D ₁	1110	no point	D ₆ D ₁	2 bit
D ₇ D ₁	1001	D ₄	D ₇ D ₄ D ₁	3 bit
D ₃ D ₂	0001	P ₀	D ₃ D ₂	2 bit
D ₄ D ₂	1111	no point	D ₄ D ₂	2 bit
D ₅ D ₂	1100	D ₇	D ₇ D ₅ D ₂	3 bit
D ₆ D ₂	1101	no point	D ₆ D ₂	2 bit
D ₇ D ₂	1010	D ₅	D ₇ D ₅ D ₂	3 bit
D ₄ D ₃	1110	no point	D ₄ D ₃	2 bit
D ₅ D ₃	1101	no point	D ₅ D ₃	2 bit
D ₆ D ₃	1100	D ₇	D ₇ D ₆ D ₃	3 bit
D ₇ D ₃	1011	D ₆	D ₇ D ₆ D ₃	3 bit
D ₅ D ₄	0011	D ₀	D ₅ D ₄ D ₀	3 bit
D ₆ D ₄	0010	P ₁	D ₆ D ₄	2 bit
D ₇ D ₄	0101	D ₁	D ₇ D ₄ D ₁	3 bit
D ₆ D ₅	0001	P ₀	D ₆ D ₅	2 bit
D ₇ D ₅	0110	D ₂	D ₇ D ₅ D ₂	3 bit
D ₇ D ₆	0111	D ₃	D ₇ D ₆ D ₃	3 bit

TABLE 6. The measured and calculated probabilities of the failure modes including of 1-, 2- and 3-bits

Error types	Number of errors measured	Probability of error	
		Measured	Theoretical
1 bit error	119/294	40.48%	39.39%
2 bits error	111/294	37.76%	37.88%
3 bits error	64/294	21.77%	22.73%

There can be several mitigation approaches if a much higher reliability is required. Periodic memory scrubbing is often used to improve the performance of the device. and a scrubbing operation will be conducted in the SRAMs exposed to heavy ions in our lab, so as to observe the relationship between the scrub-rates and the bit error rate (BER). If more redundancy is accepted, the triple-bit-correcting Golay code or the Triple Modular Redundancy (TMR) may be employed.

The research on 65 nm SRAMs may provide a reference to the manufacturers in their choice of the reinforcement model and algorithm, and to the users in their selection of device application environment and methods.

ACKNOWLEDGMENTS

The authors thank LIU Xin and ZHAO Fa-Zhan with Institute of Microelectronics, Chinese Academy of Science, for discussion on the failure modes of Hamming code, and the staff of the HIRFL accelerator, for experiment helps.

- [1] Lawrence R K and Kelly A T. IEEE Trans Nucl Sci, 2008, **55**: 3367–3374.
- [2] Heidel D F, Marshall P W, Pellish J A, *et al.* IEEE Trans Nucl Sci, 2009, **56**: 3499–3504.
- [3] Schrimpf R D, Weller R A, Mendenhall M H, *et al.* Nucl Instrum Meth B, 2007, **261**: 1133–1136.
- [4] Liu J, Duan J L, Hou M D, *et al.* Nucl Instrum Meth B, 2006, **245**: 342–345.
- [5] Bajura M A, Boughassoul Y, Naseer R, *et al.* IEEE Trans Nucl Sci, 2007, **54**: 935–945.
- [6] Radaelli D, Puchner H, Wong S, *et al.* IEEE Trans Nucl Sci, 2005, **52**: 2433–2437.
- [7] Juan Antonio Maestro and Pedro Reviriego, Study of the Effects of MBUs on the Reliability of a 150 nm SRAM Device, DAC’08

Proceedings of the 45th annual Design Automation Conference, p.930-935, California, USA, June 8–13, 2008.

- [8] Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices, JESD89A, 2006, p.10.
- [9] Palomo F R, Morilla Y, Mogollón J M, *et al.* Nucl Instrum Meth B, 2011, **269**: 2210–2216.
- [10] Nicolaïdis M. Soft errors in modern electronic systems, Germany, Springer, 2011, p.207.
- [11] Tam S. Single error correction and double error detection, Xilinx, XAPP645 (v2.2), 2006.